

**SVEUČILIŠTE U SPLITU  
FAKULTET ELEKTROTEHNIKE, STROJARSTVA I  
BRODOGRADNJE**

**SIGURNOST BEŽIČNIH MREŽA**

**BLE APLIKACIJA ZA FLIPPER ZERO**

**Jelena Matić**

**Split, veljača 2025.**

## Sadržaj

1.	UVOD .....	1
2.	OPIS TEHNOLOGIJE .....	2
2.1	Flipper Zero uređaj .....	2
2.2	Aplikacije .....	3
2.3	nRF Connect.....	3
3.	CILJEVI PROJEKTA .....	5
4.	OPIS RAZVOJA APLIKACIJE .....	6
4.1	Aplikacija za Flipper .....	6
4.2	Objašnjenje koda aplikacije .....	8
4.2.1	Uključivanje potrebnih biblioteka .....	8
4.2.2	Callback funkcije za upravljanje događajima .....	9
4.2.3	Obnavljanje stanja beacon uređaja .....	9
4.2.4	Alokacija resursa i postavljanje aplikacije .....	10
4.2.5	Čišćenje resursa .....	11
4.2.6	Pokretanje aplikacije .....	11
4.2.7	Ažuriranje stanja beacon uređaja .....	12
4.3	Podaci komunikacije .....	13
5.	ZAKLJUČAK .....	21
6.	PRILOG .....	22

# 1. UVOD

Razradom projekta napravljenog u okviru kolegija Sigurnost bežičnih mreža pojasnit ćemo način korištenja aplikacije implementirane na Flipper Zero uređaj za određivanje udaljenosti okolnih uređaja korištenjem Bluetooth Low Energy protokola. Projektom smo nastojali na jednostavan način objasniti dijelove koda koji su ključni za funkcioniranje aplikacije koja radi po principu BLE i sagledali moguće nadogradnje koda.

Rad je podijeljen na pet poglavlja. U drugom poglavlju je ukratko opisana korištena tehnologija i korišteni uređaj. Treće poglavlje donosi ciljeve projekta, a četvrto poglavlje opis razvoja projekta. Zaključne misli donesene su u istoimenom, završnom poglavlju.

## 2. OPIS TEHNOLOGIJE

### 2.1 Flipper Zero uređaj

Inovacije u području bežičnih tehnologija doprinose pojednostavljenju i boljem iskorištenju bežičnog prijenosa podataka i slanja signala bez kojih je današnji tehnološki doseg teško zamisliv. Među najnovijim inovacijama istaknut je Flipper Zero, uređaj koji predstavlja spoj naprednih funkcionalnosti s posebnim naglaskom na analizi i interakciji s bežičnim sustavima. Iako izgledom uređaj djeluje bezazleno, objedinjuje različite bežične funkcionalnosti zbog čega je ujedno i resurs koji otvara vrata istraživačkim radovima, testiranjima sigurnosti i edukaciji u području bežične sigurnosti.



*Slika 2-1 Flipper Zero uređaj*

Flipper Zero izdvaja se svojim kompaktnim dizajnom, ugrađenim senzorima, tipkovnicom i zaslonom, a korisnicima omogućava interakciju s raznim bežičnim tehnologijama na jednostavan način. U fokusu su ovog rada sposobnosti uređaja za analizu i emulaciju Bluetooth Low Energy (BLE) komunikacije. Uređaj omogućava korisnicima praćenje i analizu BLE signala, identifikaciju karakteristika uređaja, te čak i spomenutu emulaciju BLE uređaja u kontroliranom okruženju. Navedeno ostavlja prostora raznovrsnim primjenama, od testiranja i unaprjeđenja sigurnosti uređaja do istraživanja novih metoda komunikacije u IoT svijetu, a Flipper Zero već posjeduje rastuću zajednicu razvojnih programera koji njegove karakteristike iskorištavaju za kreativne i korisne open-source aplikacije dostupne svima.

## 2.2 Aplikacije

Flipper Zero uređaj koristi specifičan firmware i razvojno okruženje za prebacivanje aplikacija koji ovise najviše o programskom jeziku u kojem je aplikacija napisana. Potrebno je pripremiti razvojno okruženje. Alati koji su potrebni za rad su GNU ARM Toolchain za kompajliranje, CMake za build proces i Python za Flipper skripte. Firmware se klonira s Github-a. Aplikacije su obično pisane u C jeziku. Kod aplikacije se nalazi u file-u npr app\_name.c ,dok se entry point (ime aplikacije ,opis, stack size, ikona itd) definira u application.fam. Fam označava Flipper Application Manifest.

```
App(  
    appid="example_ble_beacon",  
    name="Example: BLE Beacon",  
    apptype=FlipperAppType.EXTERNAL,  
    entry_point="ble_beacon_app",  
    requires=["gui"],  
    stack_size=1 * 1024,  
    fap_icon="example_ble_beacon_10px.png",  
    fap_category="Examples",  
    fap_icon_assets="images",  
)
```

*Slika 2-2 application.fam*

Zatim se pokrene build proces kako se aplikacija kompajlirala. Nakon kompajliranja povežemo Flipper Zero putem USB-a i fleshamo aplikaciju pomoću fbt alata. Ako je sve dobro prošlo slijedi pokretanje aplikacije i provjera njenog ponašanja. Aplikacije moraju biti kompatibilne sa SDK-om (Software Development Kit) koji predstavlja skup alata, biblioteka i dokumentaciju koja omogućava programerima razvoj aplikacije za Flipper Zero uređaj. Korištenjem SDK i pravilnim kreiranjem app.fam Flipper Zero prepoznaje aplikaciju, učitava je i omogućava korisnicima korištenje.

## 2.3 nRF Connect

nRF Connect je mobilna aplikacija (dostupna za Android i iOS) koju je razvila kompanija Nordic Semiconductor. Koristi se za rad sa Bluetooth Low Energy (BLE) uređajima i pruža širok spektar mogućnosti za programere i inženjere koji rade sa BLE protokolom. Aplikacija može otkriti sve BLE uređaje u blizini, prikazati njihove karakteristike i identifikatore (npr. MAC adrese i UUID-ove). Omogućava pregled BLE servisa koje uređaj nudi (npr. GATT servis) i njihovih karakteristika (čitanje, pisanje, obaveštavanje itd.). Korisnici mogu ručno slati

podatke (write) ili čitati vrijednosti s BLE uređaja. Omogućava slanje HEX ili ASCII podataka. Aplikacija je odlična za otklanjanje grešaka i testiranje komunikacije s BLE uređajima.



## nRF Connect for Mobile

Nordic Semiconductor ASA

*Slika 2-3 nRF connect aplikacija*

nRF Connect omogućava simulaciju uređaja koji emitira BLE advertising (npr. testiranje beacon uređaja). Beacon (ili "svjetionik") je uređaj koji koristi Bluetooth Low Energy (BLE) tehnologiju za emitiranje malih paketa podataka u određenim vremenskim intervalima. Ovi paketi sadrže informacije koje drugi BLE uređaji, poput pametnih telefona ili tableta, mogu primiti i interpretirati. Ovi paketi su broadcast signali, što znači da ih svi BLE uređaji u dometu mogu "čuti" (nije potrebna direktna veza). Paketi obično sadrže UUID (jedinstveni identifikator uređaja), Major/Minor ID (identifikacija specifičnih beacon uređaja u grupi), RSSI (snaga signala, koja može pomoći u procjeni udaljenosti), dodatne informacije, poput senzorskih podataka (npr. temperatura, nivo baterije). Primjeri upotrebe su navigacija u zatvorenim prostorima, marketing i promocija, praćenje robe u skladištima, pomoćne tehnologije koje se koriste za asistenciju osobama s invaliditetom. Od osobite važnosti za projekt je RSSI (Received Signal Strength Indicator) koji predstavlja mjeru jačine signala koji uređaj prima od drugog uređaja putem bežične komunikacije, poput Bluetootha, Wi-Fi-ja, ili drugih radio tehnologija. U osnovi, to je pokazatelj koliko je signal jak kada stigne do prijemnika. RSSI vrijednost obično je izražena u decibelima (dBm) i ima negativnu vrijednost (što je manja, to je signal slabiji). Blizak uređaj: -30 dBm (jak signal), dok za udaljen uređaj iznosi: -90 dBm (slab signal). Jačina signala ovisi o nekoliko faktora kao što su udaljenosti između uređaja, prepreke (zidovi, namještaj) između uređaja, snaga emitiranja signala s uređaja koji ga šalje, interferencija iz okruženja. Tipične primjene RSSI-a su određivanje udaljenosti (RSSI može poslužiti za procjenu koliko je neki uređaj udaljen od izvora signala, signal je jači kada su uređaji blizu, a slabiji kada su dalje.), lokacija i praćenje (koristi se za lociranje bežičnih uređaja, kao što su beacon uređaji), monitoring kvalitete veze, IoT aplikacije. RSSI nema precizno mjerenje udaljenosti. RSSI je samo indikator jačine signala i ne daje točne podatke o udaljenosti. Na to utječu mnogi faktori, poput prepreka ili refleksije signala. Također nema stalnu vrijednost. RSSI može varirati u kratkom vremenskom intervalu zbog smetnji u okruženju.

### 3. CILJEVI PROJEKTA

Primarni je cilj projekta bio pobliže upoznavanje s BLE tehnologijom i razvoj aplikacije prilagođene razvojnom okruženju Flipper Zero uređaja kojom će biti omogućen pokušaj upravljanja komunikacijom ostvarenom BLE tehnologijom.

Sekundarni je cilj bio skladištenje podataka komunikacije, te na osnovu dobivenih podataka iščitati o kakvoj vrsti komunikacije je riječ.

Posljednji cilj bio je implementirati funkcionalnost praćenja i vizualizacije parametara BLE komunikacije u stvarnom vremenu, kako bi se olakšalo razumijevanje dinamike i ponašanja BLE uređaja u različitim uvjetima. Također, planirana je bila prezentacija napretka u dvije etape koje su pratile termine kolokvija.

## 4. OPIS RAZVOJA APLIKACIJE

### 4.1 Aplikacija za Flipper

Prateći prvotni cilj, izradu aplikacije za Flipper Zero kojom bi se, u najboljem slučaju, ostvarila komunikacija s drugim obližnjim uređajem koristeći BLE tehnologiju. Razvijena aplikacija je napisana u C programskom jeziku koja odašilje BLE beacon signale. Prvo je potrebno preuzeti službeni Flipper Zero firmware s GitHub repozitorija. To je osnovni kod koji omogućava rad Flipper Zero uređaja.

```
Microsoft Windows [Version 10.0.22631.4751]
(c) Microsoft Corporation. Sva prava pridržana.

C:\Users\frane> git clone https://github.com/flipperdevices/flipperzero-firmware.git
```

*Slika 4-1 firmware GitHub*

Zatim je potrebno ući u direktorij gdje je preuzet firmware.

```
Microsoft Windows [Version 10.0.22631.4751]
(c) Microsoft Corporation. Sva prava pridržana.

C:\Users\frane> cd flipperzero-firmware
```

*Slika 4-2 preuzimanje firmware-a*

Za kompilaciju aplikacije, koristi se Python alat pod nazivom ufbt (Unofficial Flipper Build Tool). Također, potrebno je postaviti virtualno okruženje u Pythonu kako bi svi potrebni paketi bili instalirani. Iako je aplikacija koju želimo prebaciti na Flipper Zero napisana u C jeziku, Python i virtualno okruženje se koriste za upravljanje alatima i procesima koji su potrebni za razvoj i kompilaciju tih C aplikacija. Virtualno okruženje se radi sljedećom naredbom:

```
C:\Users\frane> python -m venv flipper_env
```



Za aktivaciju virtualnog okruženja koristi se naredba:

```
C:\Users\frane> flipper_env\Scripts\activate
```

Zatim slijedi instalacija ufbt alata pomoću pip naredbe.

```
C:\Users\frane> pip install ufbt
```

Za uspješno kompajliranje aplikacije za Flipper Zero potrebno je podesiti ufbt alat. S obzirom na to da je aplikacija koja se prebacuje BLE beacon aplikacija, potrebno je pokrenuti sljedeću naredbu:

```
C:\Users\frane> ufbt APPID=example_ble_beacon
```

Ova naredba koristi ufbt alat za kompilaciju aplikacije u obliku .fap datoteke. Format datoteke .fap je specifičan format koji koristi Flipper Zero za aplikacije. On omogućuje kompajlirane aplikacije da se prenesu na uređaj i da se izvrše na samom Flipper Zero, pružajući fleksibilnost za razvoj novih funkcionalnosti i prilagodbu uređaja prema potrebama korisnika. Nakon što je aplikacija uspješno kompajlirana, ona će biti dostupna u .fap formatu. Sljedeći korak je prijenos te datoteke na Flipper Zero uređaj. Otvori se direktorij s kompajliranim aplikacijama, na primjer build/firmware, i pronađe se .fap datoteku (npr. example\_ble\_beacon.fap). Prikluči se Flipper Zero uređaj na računalo putem USB-a. Otvori se direktorij na SD kartici Flipper Zero uređaja (u direktoriju /apps/). Za kraj se kopira .fap datoteku u direktorij /apps/ na SD kartici Flipper Zero.

Na Flipper Zero, nakon što je aplikacija kopirana na SD karticu, odabere se aplikacija iz izbornika i pokrene se. Aplikacija će početi emitirati BLE beacon signal na temelju postavki definiranim u aplikaciji.



*Slika 4-3 Izgled aplikacije*

## 4.2 Objašnjenje koda aplikacije

Aplikacija koja se koristi za odašiljanje BLE beacon signala na Flipper Zero uređaju razvijena je u C programskom jeziku. Glavni cilj aplikacije je omogućiti Flipper Zero uređaju da odašilje specifičan beacon signal koristeći BLE (Bluetooth Low Energy) tehnologiju. Sljedeće objašnjenje detaljno razrađuje funkcionalnosti unutar glavnih dijelova koda.

### 4.2.1 Uključivanje potrebnih biblioteka

Na početku koda uključuju se osnovne biblioteke i zaglavlja koja omogućuju korištenje funkcionalnosti potrebnih za BLE komunikaciju i interakciju s Flipper Zero uređajem:

```
1 #include "ble_beacon_app.h"
2 #include <extra_beacon.h>
3 #include <furi_hal_version.h>
4 #include <string.h>
```

*Slika 4-4 Uključene biblioteke*

- **ble\_beacon\_app.h:** Glavni header za aplikaciju koji definira strukture i funkcije specifične za BLE beacon aplikaciju.
- **extra\_beacon.h:** Sadrži funkcije za konfiguraciju i upravljanje BLE beacon signalom.
- **furi\_hal\_version.h:** Omogućuje pristup verziji i informacijama o hardveru Flipper Zero uređaja.
- **string.h:** Standardna biblioteka koja omogućuje rad s nizovima znakova (stringovima), što je korisno za manipulaciju podacima beacon signala.

### 4.2.2 Callback funkcije za upravljanje događajima

Kod aplikacije sadrži nekoliko callback funkcija koje upravljaju različitim događajima unutar aplikacije.

```
8 - static bool ble_beacon_app_custom_event_callback(void* context, uint32_t
    event) {
9     furi_assert(context);
10    BleBeaconApp* app = context;
11    return scene_manager_handle_custom_event(app->scene_manager, event);
12 }
13
14 - static bool ble_beacon_app_back_event_callback(void* context) {
15     furi_assert(context);
16     BleBeaconApp* app = context;
17     return scene_manager_handle_back_event(app->scene_manager);
18 }
19
20 - static void ble_beacon_app_tick_event_callback(void* context) {
21     furi_assert(context);
22     BleBeaconApp* app = context;
23     scene_manager_handle_tick_event(app->scene_manager);
24 }
```

*Slika 4-5 Callback funkcije*

- **ble\_beacon\_app\_custom\_event\_callback:** Ova funkcija obrađuje prilagođene događaje koji dolaze iz različitih scena aplikacije. Koristi `scene_manager_handle_custom_event` za upravljanje događajem.
- **ble\_beacon\_app\_back\_event\_callback:** Funkcija koja se poziva kad korisnik želi izaći iz trenutne scene. Upotrebljava `scene_manager_handle_back_event` kako bi se vratio na prethodnu scenu.
- **ble\_beacon\_app\_tick\_event\_callback:** Ova funkcija se poziva periodički (svakih 100 ms u ovom slučaju) za obavljanje redovnih aktivnosti, kao što je ažuriranje prikaza ili provjera stanja.

### 4.2.3 Obnavljanje stanja beacon uređaja

Funkcija **ble\_beacon\_app\_restore\_beacon\_state** vraća prethodne postavke beacon uređaja, ako su bile spremljene:

```

26 static void ble_beacon_app_restore_beacon_state(BleBeaconApp* app) {
27     // Restore beacon data from service
28     GapExtraBeaconConfig* local_config = &app->beacon_config;
29     const GapExtraBeaconConfig* config = furi_hal_bt_extra_beacon_get_config();
30     if(config) {
31         // We have a config, copy it
32         memcpy(local_config, config, sizeof(app->beacon_config));
33     } else {
34         local_config->min_adv_interval_ms = 50;
35         local_config->max_adv_interval_ms = 150;
36         local_config->adv_channel_map = GapAdvChannelMapAll;
37         local_config->adv_power_level = GapAdvPowerLevel_0dBm;
38         local_config->address_type = GapAddressTypePublic;
39         memcpy(
40             local_config->address, furi_hal_version_get_ble_mac(), sizeof
                (local_config->address));
41         local_config->address[0] ^= 0xFF;
42         local_config->address[3] ^= 0xFF;
43
44         furi_check(furi_hal_bt_extra_beacon_set_config(local_config));
45     }
46     app->is_beacon_active = furi_hal_bt_extra_beacon_is_active();
47     app->beacon_data_len = furi_hal_bt_extra_beacon_get_data(app->beacon_data);
48 }

```

*Slika 4-6 Funkcije obnavljanja stanja*

- Funkcija najprije pokušava preuzeti prethodnu konfiguraciju beacon uređaja.
- Ako konfiguracija nije pronađena, postavlja se osnovna konfiguracija poput intervala oglašavanja, snage i MAC adrese.
- Nakon toga, dohvaća se i sprema trenutni status beacon uređaja.

#### 4.2.4 Alokacija resursa i postavljanje aplikacije

Funkcija **ble\_beacon\_app\_alloc** alocira potrebne resurse za aplikaciju, uključujući GUI (grafičko korisničko sučelje), scene, i upravljače prikazom:

- Aplikacija alocira sve potrebne komponente kao što su GUI, scena i dispatcher za upravljanje prikazom na ekranu.
- Dodaju se različite vrste prikaza, uključujući submenu, dijaloške prozore i unos bajtova.

### 4.2.5 Čišćenje resursa

Funkcija **ble\_beacon\_app\_free** uklanja sve alocirane resurse i zatvara otvorene resurse, kao što su GUI i druge komponente aplikacije:

```
87 - static void ble_beacon_app_free(BleBeaconApp* app) {
88     view_dispatcher_remove_view(app->view_dispatcher, BleBeaconAppViewByteInput);
89     view_dispatcher_remove_view(app->view_dispatcher, BleBeaconAppViewSubmenu);
90     view_dispatcher_remove_view(app->view_dispatcher, BleBeaconAppViewDialog);
91
92     free(app->byte_input);
93     free(app->submenu);
94     free(app->dialog_ex);
95
96     free(app->scene_manager);
97     free(app->view_dispatcher);
98
99     free(app->status_string);
100
101     furi_record_close(RECORD_NOTIFICATION);
102     furi_record_close(RECORD_GUI);
103     app->gui = NULL;
104
105     free(app);
106 }
```

*Slika 4-7 Čišćenje resursa*

### 4.2.6 Pokretanje aplikacije

Funkcija **ble\_beacon\_app** pokreće aplikaciju i upravlja cijelim ciklusom rada aplikacije, od inicijalizacije resursa do pokretanja beacon signala:

```

108 int32_t ble_beacon_app(void* args) {
109     UNUSED(args);
110
111     BleBeaconApp* app = ble_beacon_app_alloc();
112
113     scene_manager_next_scene(app->scene_manager, BleBeaconAppSceneRunBeacon);
114
115     view_dispatcher_run(app->view_dispatcher);
116
117     ble_beacon_app_free(app);
118     return 0;
119 }

```

*Slika 4-8 Pokretanje aplikacije*

#### 4.2.7 Ažuriranje stanja beacon uređaja

Funkcija **ble\_beacon\_app\_update\_state** ažurira stanje beacon uređaja, postavlja konfiguraciju beacon podataka, i pokreće beacon ako je aktivan:

```

121 void ble_beacon_app_update_state(BleBeaconApp* app) {
122     furi_hal_bt_extra_beacon_stop();
123
124     furi_check(furi_hal_bt_extra_beacon_set_config(&app->beacon_config));
125
126     app->beacon_data_len = 0;
127     while((app->beacon_data[app->beacon_data_len] != 0) &&
128         (app->beacon_data_len < sizeof(app->beacon_data))) {
129         app->beacon_data_len++;
130     }
131
132     FURI_LOG_I(TAG, "beacon_data_len: %d", app->beacon_data_len);
133
134     furi_check(furi_hal_bt_extra_beacon_set_data(app->beacon_data, app
        ->beacon_data_len));
135
136     if(app->is_beacon_active) {
137         furi_check(furi_hal_bt_extra_beacon_start());
138     }

```

*Slika 4-9 Ažuriranje beacon uređaja*

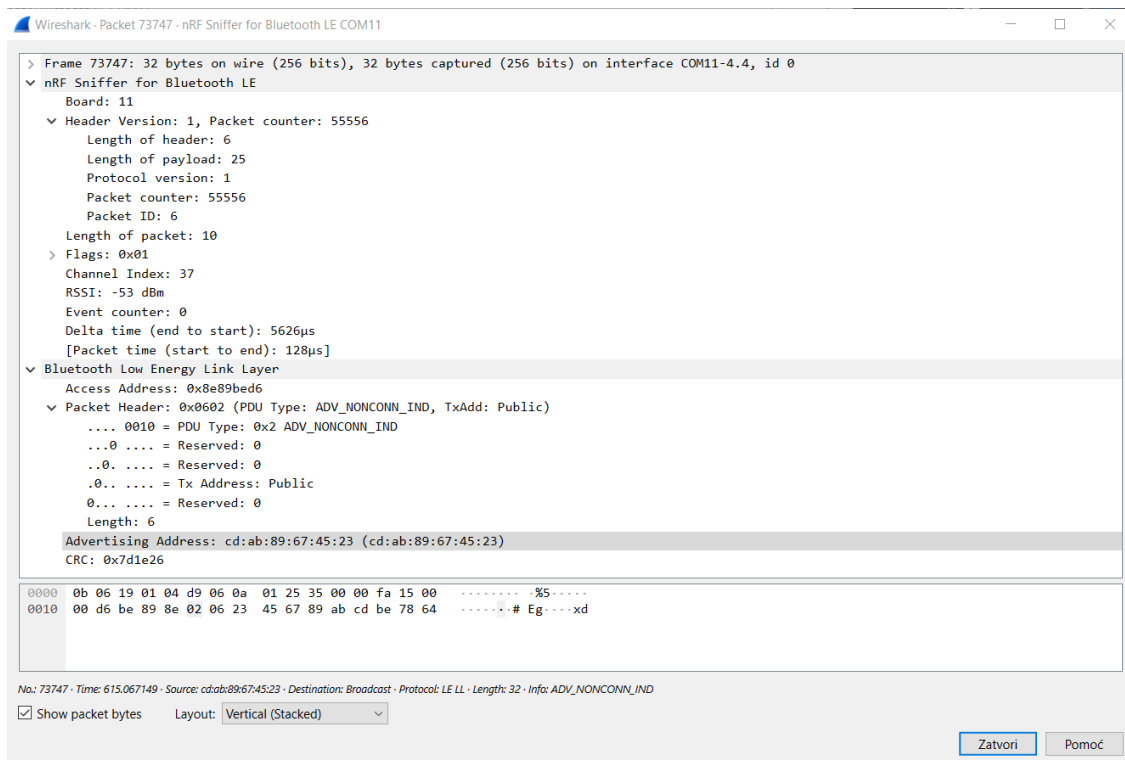
## 4.3 Podaci komunikacije

Za testiranje BLE beacon aplikacije, provedeno je mjerenje RSSI (Received Signal Strength Indicator) signala na različitim udaljenostima od Flipper Zero uređaja. Ovaj postupak omogućuje praćenje jačine signala na temelju udaljenosti, što je ključno za analizu učinkovitosti i dosega beacon signala. Podaci su prikupljeni pomoću mobilne aplikacije nRF Connect, koja omogućava precizno praćenje i mjerenje različitih BLE parametara, uključujući RSSI signal. Također je korišten i WireShark program u kombinaciji sa snifferom Adafruit. WireShark je moćan alat za analizu mrežnog prometa koji omogućava detaljno praćenje i pregledanje svih paketa koji prolaze kroz mrežu. U kombinaciji sa snifferom, poput Adafruit Bluefruit LE Sniffer, omogućava hvatanje specifičnih Bluetooth Low Energy (BLE) komunikacija. Adafruit sniffer presreće BLE promet između uređaja, dok WireShark analizira i dekodira pakete u čitljivom formatu.



*Slika 4-10 Sniffer Adafruit*

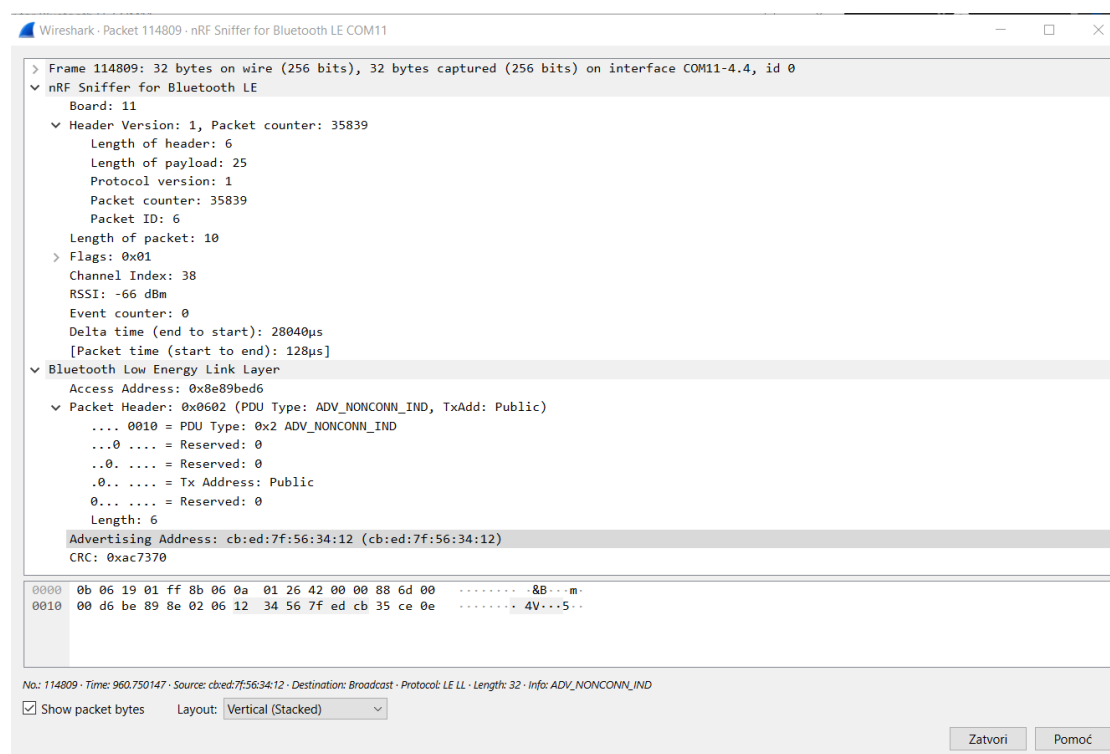
Rezultati ukazuju da je sniffer uhvatio pakete koje šalje Flipper Zero. Ovaj paket je BLE reklamni (advertising) paket koji je sniffer uhvatio na kanalu 37. Uređaj s adresom cd:ab:89:67:45:23 šalje ne-konektivnu reklamu s dobrim signalom (RSSI -53 dBm). Ovo je tipično za uređaje poput senzora koji samo emitiraju informacije, bez omogućavanja veze s drugim uređajima.



*Slika 4-11 Rezultat sniffera*

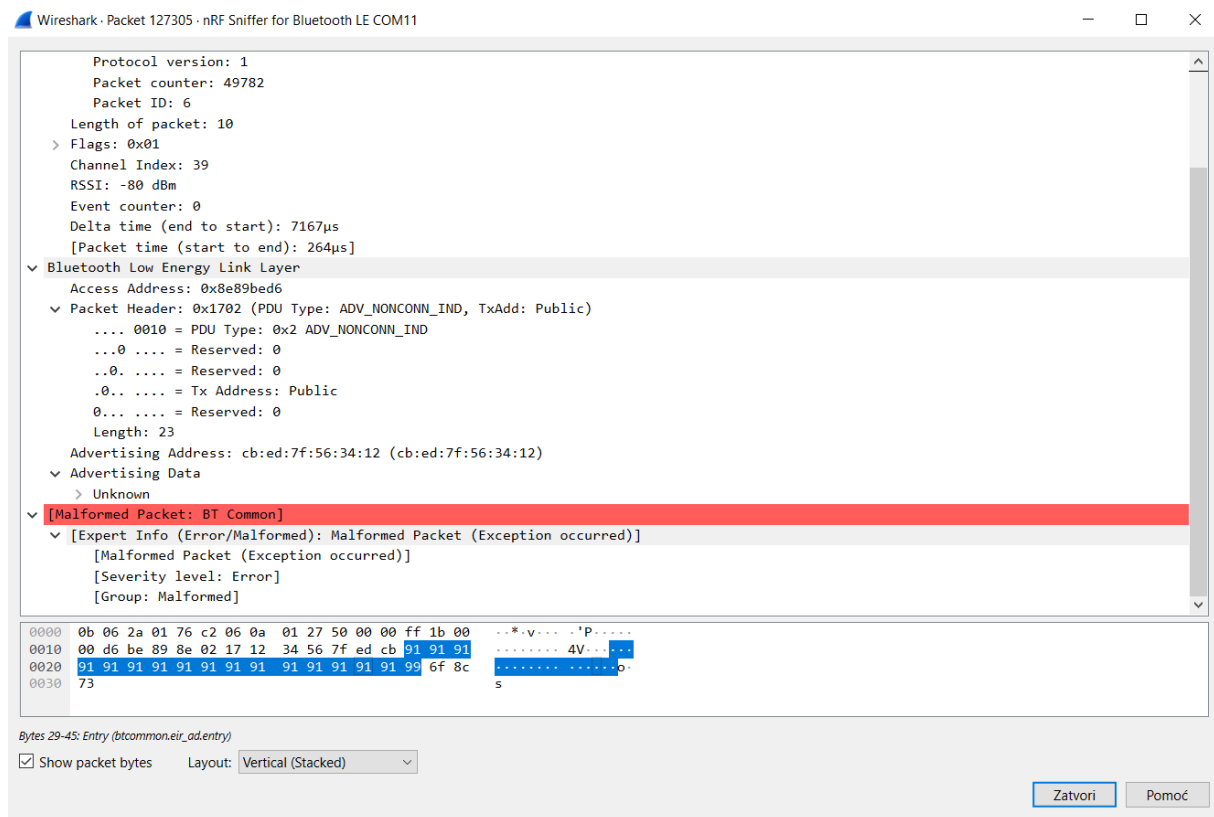
Sljedeći korak je bila ručna promjena mac adrese da se provjeri mogućnost wiresharka da prepozna da li je u pitanju isti uređaj od maloprije ili ne. Wireshark ga nije prepoznao, tretirao ga je kao novi uređaj s MAC adresom cb:ed:7f:56:34:12 koji šalje pakete na kanalu 38. Snaga signala je -66 dBm, što ukazuje na stabilan, ali ne savršen signal. Tip paketa (ADV\_NONCONN\_IND) sugerira da uređaj reklamira svoju prisutnost, ali nije namenjen uspostavljanju veze s drugim uređajima.





Slika 4-12 Promjena mac adrese

Zbog promjene MAC adrese slijedio je pokušaj promjene sadržaja paketa, no to nije bilo moguće. Paket je BLE reklamni paket koji je poslan na kanal 39 i ima MAC adresu cb:ed:7f:56:34:12. Signal je -80 dBm, što ukazuje na slabiji signal. Paket je označen kao *malformiran* jer sadrži grešku u strukturi podataka.



*Slika 4-13 Promjena sadržaja paketa*

Nakon prikupljanja podataka, isti su analizirani korištenjem Python programskog jezika, koji je omogućio daljnje statističke obrade i vizualizaciju podataka.

```
PS C:\Users\Bruno\OneDrive\Radna površina\sbn> py procjena.py
Unesite RSSI vrijednost (dBm): -71
Predviđena udaljenost: 1.68 metara
PS C:\Users\Bruno\OneDrive\Radna površina\sbn> 
```

*Slika 4-14 Python datoteka*

Podaci o RSSI signalu pohranjeni su u CSV datotekama, gdje svaka datoteka odgovara određenoj udaljenosti od izvora beacon signala. Datoteke s podacima su označene prema sljedećim udaljenostima:

- 0 metara.csv – Podaci prikupljeni na udaljenosti od 0 metara (tj. neposredno uz beacon uređaj).

1	Timestamp	Source	Level	Line
2	12:12:00.4	RSSI	Normal	-44 dBm
3	12:12:07.4	RSSI	Normal	-40 dBm
4	12:12:09.2	RSSI	Normal	-41 dBm
5	12:12:12.7	RSSI	Normal	-39 dBm
6	12:12:14.4	RSSI	Normal	-39 dBm
7	12:12:19.7	RSSI	Normal	-40 dBm
8	12:12:21.8	RSSI	Normal	-39 dBm
9	12:12:23.6	RSSI	Normal	-41 dBm
10	12:12:28.8	RSSI	Normal	-38 dBm
11	12:12:35.9	RSSI	Normal	-38 dBm
12	12:12:37.6	RSSI	Normal	-39 dBm
13	12:12:42.9	RSSI	Normal	-38 dBm
14	12:12:44.6	RSSI	Normal	-37 dBm
15	12:12:51.6	RSSI	Normal	-38 dBm
16	12:12:53.4	RSSI	Normal	-42 dBm
17	12:12:58.7	RSSI	Normal	-37 dBm

*Slika 4-15 Podaci 0 metara*

- 1 metar.csv – Podaci prikupljeni na udaljenosti od 1 metra.

1	Timestamp	Source	Level	Line
2	12:16:25.3	RSSI	Normal	-65 dBm
3	12:16:27.1	RSSI	Normal	-78 dBm
4	12:16:28.8	RSSI	Normal	-67 dBm
5	12:16:32.3	RSSI	Normal	-68 dBm
6	12:16:34.1	RSSI	Normal	-65 dBm
7	12:16:35.9	RSSI	Normal	-68 dBm
8	12:16:37.6	RSSI	Normal	-74 dBm
9	12:16:39.4	RSSI	Normal	-77 dBm
10	12:16:41.1	RSSI	Normal	-63 dBm
11	12:16:41.1	RSSI	Normal	-63 dBm
12	12:16:42.9	RSSI	Normal	-63 dBm
13	12:16:44.6	RSSI	Normal	-62 dBm
14	12:16:46.4	RSSI	Normal	-63 dBm
15	12:16:53.4	RSSI	Normal	-64 dBm
16	12:16:58.7	RSSI	Normal	-62 dBm
17	12:17:03.9	RSSI	Normal	-62 dBm
18	12:17:03.9	RSSI	Normal	-62 dBm
19	12:17:16.2	RSSI	Normal	-63 dBm
20	12:17:18.0	RSSI	Normal	-62 dBm
21	12:17:19.7	RSSI	Normal	-64 dBm
22	12:17:21.8	RSSI	Normal	-62 dBm

*Slika 4-16 Podaci 1 metar*

- 3 metra.csv – Podaci prikupljeni na udaljenosti od 3 metra.

1	Timestamp	Source	Level	Line
2	12:19:39.4	RSSI	Normal	-70 dBm
3	12:19:41.1	RSSI	Normal	-73 dBm
4	12:19:46.4	RSSI	Normal	-68 dBm
5	12:19:48.1	RSSI	Normal	-65 dBm
6	12:19:49.9	RSSI	Normal	-64 dBm
7	12:19:58.7	RSSI	Normal	-79 dBm
8	12:20:09.2	RSSI	Normal	-76 dBm
9	12:20:12.7	RSSI	Normal	-78 dBm
10	12:20:18.0	RSSI	Normal	-91 dBm
11	12:20:20.1	RSSI	Normal	-74 dBm
12	12:20:20.1	RSSI	Normal	-74 dBm
13	12:20:27.1	RSSI	Normal	-74 dBm
14	12:20:28.8	RSSI	Normal	-90 dBm
15	12:20:32.3	RSSI	Normal	-83 dBm
16	12:20:34.1	RSSI	Normal	-75 dBm
17	12:20:34.1	RSSI	Normal	-75 dBm
18	12:20:41.1	RSSI	Normal	-91 dBm
19	12:20:49.9	RSSI	Normal	-74 dBm
20	12:20:51.6	RSSI	Normal	-91 dBm
21	12:20:58.7	RSSI	Normal	-92 dBm
22	12:21:03.6	RSSI	Normal	-74 dBm

*Slika 4-17 Podaci 3 metra*

- 5 metara.csv – Podaci prikupljeni na udaljenosti od 5 metara.

1	Timestamp	Source	Level	Line
2	12:23:14.5	RSSI	Normal	-82 dBm
3	12:23:19.8	RSSI	Normal	-89 dBm
4	12:23:20.1	RSSI	Normal	-84 dBm
5	12:23:23.6	RSSI	Normal	-84 dBm
6	12:23:28.8	RSSI	Normal	-89 dBm
7	12:23:30.6	RSSI	Normal	-83 dBm
8	12:23:32.3	RSSI	Normal	-82 dBm
9	12:23:34.1	RSSI	Normal	-84 dBm
10	12:23:34.1	RSSI	Normal	-86 dBm
11	12:23:42.9	RSSI	Normal	-85 dBm
12	12:23:44.6	RSSI	Normal	-87 dBm
13	12:23:44.6	RSSI	Normal	-86 dBm
14	12:23:49.9	RSSI	Normal	-80 dBm
15	12:23:58.7	RSSI	Normal	-80 dBm
16	12:24:03.9	RSSI	Normal	-82 dBm
17	12:24:07.5	RSSI	Normal	-89 dBm
18	12:24:16.2	RSSI	Normal	-77 dBm
19	12:24:21.8	RSSI	Normal	-92 dBm
20	12:24:23.6	RSSI	Normal	-86 dBm
21	12:24:28.9	RSSI	Normal	-83 dBm
22	12:24:32.4	RSSI	Normal	-82 dBm
23	12:24:34.1	RSSI	Normal	-84 dBm

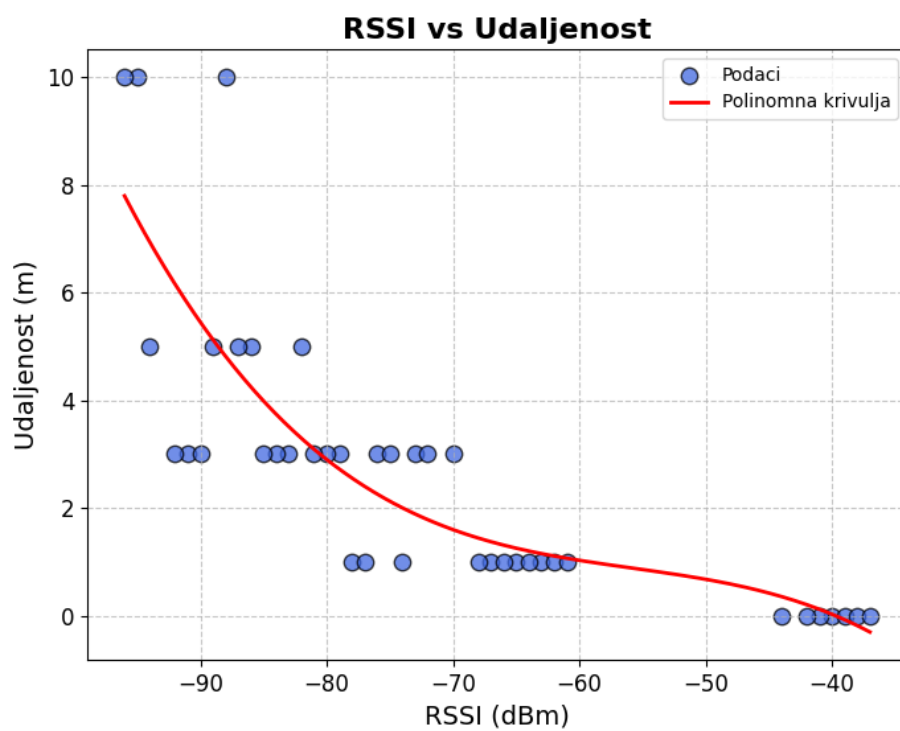
*Slika 4-18 Podaci 5 metara*

- 10 metara.csv – Podaci prikupljeni na udaljenosti od 10 metara.

1	Timestamp	Source	Level	Line
2	12:26:44.7	RSSI	Normal	-76 dBm
3	12:26:44.7	RSSI	Normal	-76 dBm
4	12:26:46.4	RSSI	Normal	-90 dBm
5	12:26:49.9	RSSI	Normal	-89 dBm
6	12:26:53.4	RSSI	Normal	-86 dBm
7	12:26:56.9	RSSI	Normal	-87 dBm
8	12:26:58.7	RSSI	Normal	-88 dBm
9	12:27:02.2	RSSI	Normal	-90 dBm
10	12:27:05.7	RSSI	Normal	-91 dBm
11	12:27:07.5	RSSI	Normal	-85 dBm
12	12:27:14.5	RSSI	Normal	-87 dBm
13	12:27:27.1	RSSI	Normal	-85 dBm
14	12:27:27.1	RSSI	Normal	-84 dBm
15	12:27:55.2	RSSI	Normal	-89 dBm
16	12:28:03.9	RSSI	Normal	-90 dBm
17	12:28:05.7	RSSI	Normal	-90 dBm
18	12:28:11.0	RSSI	Normal	-92 dBm
19	12:28:12.7	RSSI	Normal	-87 dBm
20	12:28:18.0	RSSI	Normal	-92 dBm
21	12:28:19.7	RSSI	Normal	-88 dBm
22	12:28:20.1	RSSI	Normal	-88 dBm

*Slika 4-19 Podaci 10 metara*

Svaka od ovih CSV datoteka sadrži mjerenja RSSI signala, koja su potom analizirana kako bi se dobio uvid u ponašanje beacon signala na različitim udaljenostima. Na osnovu dobivenih podataka prikazana je polinomna krivulja koja povezuje RSSI i udaljenost. RSSI signal obično opada s udaljenosti, a u idealnim uvjetima (bez smetnji i prepreka) može slijediti neki oblik zakrivljene funkcije.



*Slika 4-20 Graf podataka*

U mnogim slučajevima, opadanje RSSI-ja s udaljenostima može biti vrlo složeno, pa je često potrebno koristiti polinomske funkcije višeg stupnja kako bi se točno modelirao pad signala. U praksi, RSSI i udaljenost često ne slijede jednostavnu linearnu (pravocrtnu) funkciju.

## 5. ZAKLJUČAK

Projektom "BLE Aplikacija za Flipper Zero" ostvarena je uspješna demonstracija primjene Bluetooth Low Energy (BLE) tehnologije za odašiljanje beacon signala te analiza njihovih karakteristika na različitim udaljenostima. Kroz razvoj aplikacije u programskom jeziku C te implementaciju na Flipper Zero uređaju, detaljno su proučeni procesi konfiguracije, razvoja i testiranja BLE aplikacija. Mjerenjem RSSI signala na različitim udaljenostima, korištenjem mobilne aplikacije nRF Connect, omogućeno je bolje razumijevanje ponašanja BLE signala u kontroliranim uvjetima. Analizom prikupljenih podataka i vizualizacijom polinomne krivulje koja povezuje RSSI i udaljenost potvrđeno je da jačina signala opada s udaljenošću, pri čemu su na rezultate utjecali faktori poput smetnji i prepreka u okruženju.

Primjenom ovog projekta demonstrirana je fleksibilnost Flipper Zero uređaja u razvoju aplikacija za bežičnu komunikaciju, kao i važnost BLE tehnologije u modernim IoT sustavima. Nadalje, identificirane su mogućnosti za daljnji razvoj, poput preciznijeg modeliranja RSSI-udaljenost odnosa, poboljšanja grafičkog korisničkog sučelja aplikacije te proširenja funkcionalnosti za širu primjenu u analizi BLE signala.

Ovaj rad pokazao je kako inovativna tehnologija poput Flipper Zero može biti ključan alat za istraživanje, testiranje i edukaciju u području bežičnih mreža, čime se otvaraju mogućnosti za buduće projekte u domeni sigurnosti bežičnih sustava i njihove optimizacije.

## 6. PRILOG

Slika 2-1 Flipper Zero uređaj .....	2
Slika 2-2 application.fam .....	3
Slika 2-3 nRF connect aplikacija .....	4
Slika 4-1 firmware GitHub.....	6
Slika 4-2 preuzimanje firmware-a.....	6
Slika 4-3 Izgled aplikacije.....	7
Slika 4-4 Uključene biblioteke.....	8
Slika 4-5 Callback funkcije .....	9
Slika 4-6 Funkcije obnavljanja stanja .....	10
Slika 4-7 Čišćenje resursa .....	11
Slika 4-8 Pokretanje aplikacije.....	12
Slika 4-9 Ažuriranje beacon uređaja .....	12
Slika 4-10 Sniffer Adafruit.....	13
Slika 4-11 Rezultat sniffera.....	14
Slika 4-12 Promjena mac adrese .....	15
Slika 4-13 Promjena sadržaja paketa .....	16
Slika 4-14 Python datoteka .....	16
Slika 4-15 Podaci 0 metara.....	17
Slika 4-16 Podaci 1 metar .....	17
Slika 4-17 Podaci 3 metra .....	18
Slika 4-18 Podaci 5 metara.....	18
Slika 4-19 Podaci 10 metara.....	19
Slika 4-20 Graf podataka.....	20